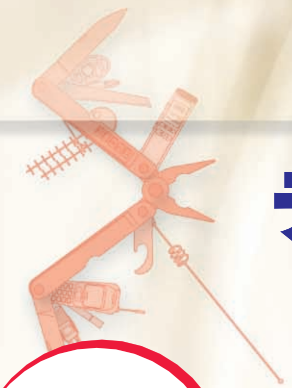


ディジタル信号処理とFPGAの橋渡しをするツールを利用

Appendix

森田 康

ここでは，本特集の設計事例で利用されている米国Xilinx社の「System Generator for DSP」について解説する．System Generator for DSPは，MATLAB/Simulink上でFPGA用の機能マクロを利用したり，SimulinkのモデルをFPGA用のHDLデータやテストベンチに変換するためのツールである．MATLAB/Simulinkのシミュレータ機能と実ハードウェア（FPGA）をリンクさせて，協調シミュレーションを行う機能も備えている．（編集部）

米国Xilinx社のSystem Generator for DSP（以下，System Generator）は，FPGA開発専用の高位設計ツールです．従来のFPGA設計は，仕様書をもとにHDLでコーディングするところから作業が始まります．一方，System Generatorによる設計では，システムをブロック線図形式で記述するところから始まります．

ディジタル信号処理システムや通信システムのブロック線図の入力やシミュレーションには，米国The MathWorks社の「MATLAB/Simulink」がよく利用されています．System GeneratorはこのMATLAB/Simulinkとともに使用します．ユーザから見ると，System GeneratorはMATLAB/Simulinkの拡張機能のように扱えます．

System Generatorの特徴は，ブロック線図のモデルから等価

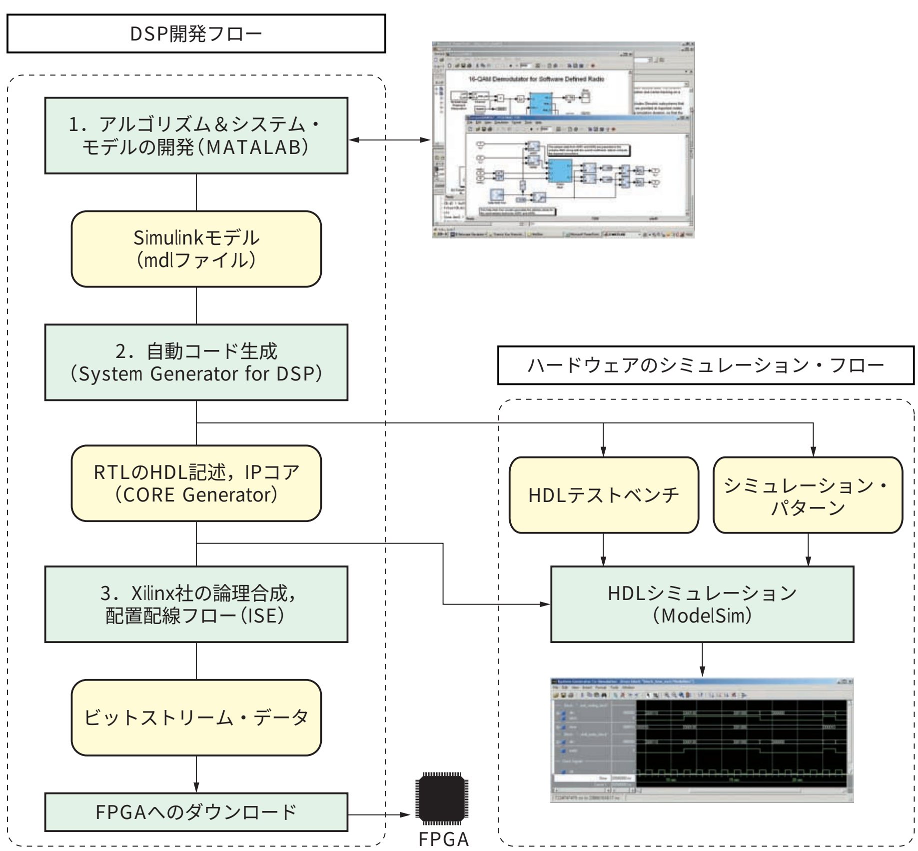


図1
System Generatorによる設計フロー
MATLAB/Simulinkを用いてブロック線図表現でアルゴリズムを記述する．アルゴリズムの動作に問題がないことを確認した後，ブロック線図の構成要素をSystem Generatorが提供するXilinx Blocksetを使ってモデリングする．再度シミュレーションを行い，量子化の悪影響が出ていないかなどをチェックする．動作に問題がなければ，このモデルと等価なHDLコードを生成する．

なHDLコードを生成することにあります．このとき，FPGAのハード・マクロやIPコア，そのほかのリソースを有効活用したHDLコードを生成するという特徴があります（下掲のコラム「System GeneratorはVirtex-4搭載XtremeDSPスライスに対応」を参照）．

● MATLAB/SimulinkとFPGA開発環境をリンク

System Generatorを利用したFPGA設計のフローを図1に示します．

まず，MATLAB/Simulinkを用いてブロック線図表現でアルゴリズムを記述します．これがリファレンス・モデルとなります．

次に，動作検証によってアルゴリズムの動作に問題がないことを確認した後，FPGAに実装するモデルをSystem Generatorが提供するXilinx Blockset（FPGA向けの機能ブロック・ライブラリ）のブロックで構築していきます．このとき，リファレンス・モデルが浮動小数点モデルであっても，一般にFPGAに実装するモデルは固定小数点で作成する必要があります．そのため，再度シミュレーションを行い，量子化の悪影響が出ていないかなどをチェックします．

固定小数点モデルの動作に問題がなければ，System Generatorの機能を使ってこのモデルと等価なHDLコードを生成します．その際，一部のブロックは「CORE Generator」というXilinx社のIPコア生成ツールを使って生成します．また，HDLコードを生成するときに，テストベンチを生成することも可能です．

● アルゴリズム検討や実装設計に使われている

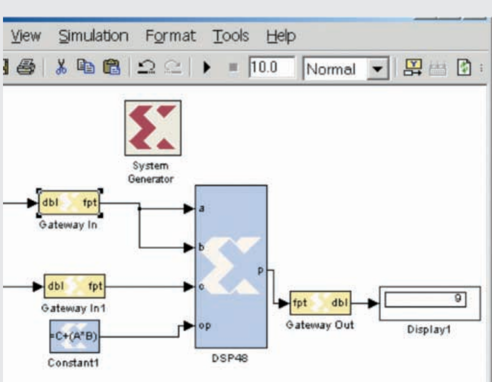
System Generatorはディジタル信号処理アルゴリズムの検討に使われる場合もあれば，実際のハードウェア設計（部分設計やシステム全体の設計）に使われる場合もあります（右掲のコラム「Simulinkモデルが『実行可能な仕様書』になる」を参照）．

1）アルゴリズムの検討

アルゴリズムを検討する際には，MATLAB/Simulinkでシステム・レベルのシミュレーションを行った後，System Generatorのリソース評価機能を利用して実装時の回路規模やメモリ容量を見積もります．また，ハードウェア協調シミュレーション（いわゆるコシミュレーション）の機能を備えており，これを利用して検証の効率化を図ることができます．協調シミュレーションとは，MATLAB/Simulinkのシミュレーション機能を外部の実ハードウェア（FPGA）と連携動作させて，アルゴリズムの中の一部のブロックの処理を実ハードウェアに肩代わりさせる手法です．こうすることで，シミュレーションの動作と実ハードウェアの動作を容易に比較できます．また，シミュレーション速度の向上も期待できます（実ハードウェアが一種のエミュレータとして機能する）．

COLUMN System GeneratorはVirtex-4搭載XtremeDSPに対応

Virtex-4では，ディジタル信号処理システムの実装に有効な「XtremeDSPスライス」が搭載されています．このスライスは，I/Oレジスタ，18×18ビットの符号付き乗算器，3入力加減算器，Op-mode付きマルチプレクサを内蔵しています．ここでOp-modeとは，Xtreme DSPスライスの内部リソース間の接続を動的に切り替える機能（いわば，非常に小さなダイナミック・コンフィグレーション機能）です．また，XtremeDSPスライスどうしを直結する専用ポート，配線リソースも用意されています．



図A-1　Op-mode対応DSP48ブロック

XtremeDSPスライスのI/Oレジスタは，ほかのハードウェア・リソースを消費せずに，データシート上で示される最高速で動作します．また，フィルタ構造を実現する際に，性能低下の原因となるアダー・ツリー構造ではなく，アダー・チェイン構造（積和演算器をカスケードにつなげた構造）を構成できます．

今回，XtremeDSPスライスにOp-mode付きマルチプレクサを導入したことにより，一つのスライスで40種類以上の機能を実現できるようになりました．また，その設定を1クロック周期の間に動的に変更することも可能です．

本誌付属のDVD-ROMに収録されているSystem Generator for DSP v6.3には，XtremeDSPスライスに対応したブロックが用意されています．Op-modeを入力する専用ポートとモード指定のための定数をConstantブロックで与えることで，Op-modeをブロックの外部入力として利用できるようになります（図A-1）．

2）ブロック・レベル設計（部分設計）

ブロック・レベルの設計を行う際には，System Generatorを使ってHDLラッパ（wrapper）を生成します．これを利用することで，ブロック外部とのインターフェース部について，タイミング設定などの細かい作業を省略することができます．

3）システム全体の設計

システム全体の設計に利用する際には，System Generatorを利用してMATLAB/SimulinkのモデルをHDLデータに変換します．このとき，System Generatorによって生成されるファイルを表1に示します．

● MATLABのオプションをインストール

System Generatorを動作させるには，以下のソフトウェアがあらかじめインストールされている必要があります．

- MATLAB v7.0x/Simulink v6.0（R14），またはMATLAB v6.5.1/Simulink v5.1（R13.1）
- Signal Processing Toolbox，Filter Design Toolbox，Communications Toolbox，Signal Processing Blockset，Communications BlocksetといったMATLABのオプション
- Xilinx ISE Foundation（または同Alliance）6.3i，および同Service Pack 2以上
- ModelSim XE（Xilinx Edition）/PE/SE v5.8d以降（協調シミュレーションを行う場合）

インストール方法の詳細については，本誌付属のDVD-ROMに収録されたインストール・ガイドを参照してください．

● MATLABでフィルタを設計する

以降では，ディジタルFIR（finite impulse response）フィルタ

表1　System Generatorが生成するファイル

ファイル名と属性	内　容
<design>_files.vhd/.v	システムが記述されたHDLファイル
<design>_clk_wrapper.vhd/.v	<design>_files.vhd/.vのHDLのラッパ．クロックとクロック・イネーブルをドライブする
conv_pkg.vhd/.v	<design>_files.vhd/.vで使われる定数およびファンクションが記述されている
.edn	HDLを生成する際，System Generatorは設計の一部に対してCORE Generatorを適用する．これによって生成されたファイルには.ednという拡張子が付く
globals	設計を表すキー/値の情報を含み，Perl hash tableという形式で構成される．キー/値はPerlスクリプトによって解釈される
<design>.xcf（or .ncf）	タイミングとポート位置情報を含む．このファイルはXilinx社の論理合成ツール（XST）と配置配線ツールで使用される．XST以外の論理合成ツールを使用する場合は，拡張子が.ncfになる
<design>.npl	HDLデータとEDIFネットリストをXilinx社のプロジェクト管理ツール（Project Navigator）内で利用するためのプロジェクト・ファイル
hdlFiles	System Generatorによって生成されたすべてのHDLファイルのリスト．HDLの依存関係を一覧できる
synplify_<design>.prj，xst_<design>.prj，spectrum_<design>.tcl	HDLデータを論理合成ツール（Synplify，XST，Spectrum）によってコンパイルするときに使用されるファイル
vcom.do	生成されたHDLをModelSimでシミュレーションする際に使用されるファイル
sysgen.log，postnetlist.log	System Generatorの変換過程をレポートするログ
各種インターフェース・ファイル，interface.txt	System Generatorは変換結果をほかのツールへ渡すためのインターフェース・ファイルを生成する．interface.txtにはテキスト形式で同様の情報が出力される
<design>_config.m	コンフィグレーション・ファイル．System GeneratorのブラックボックスにHDLデータやEDIFネットリストを割り当てるために使用する．詳細については，Xilinx社のユーザ・ガイドを参照

COLUMN

Simulinkモデルが「実行可能な仕様書」になる

今日のディジタル通信システムの開発では，アルゴリズムの複雑化や大規模化が進んでいます．開発期間が短くなる一方で，設計・検証作業の負荷が増大しています．

こうした問題を解消するために，MATLAB/Simulinkを導入するユーザが増えています．ユーザは，MATLAB/Simulinkの環境を使ってアルゴリズムを開発したり，システム全体の検証やプロトタイピング・モデルの作成を行ったりしています．

MATLAB/Simulinkはブロック線図という直感的にわかりやすい形式でシステムやアルゴリズムを表現します．また，シミュレーション結果も簡単にグラフ化することができるため，多くの設計者がSimulinkモデルを「実行可能な仕様書（executable specification）」として利用しています．Simulinkモデルを仕様書とすることで，仕様書そのもののバグや，仕様書の解釈に起因する後工程（実装設計時など）のトラブルを減らすことが期待できます．

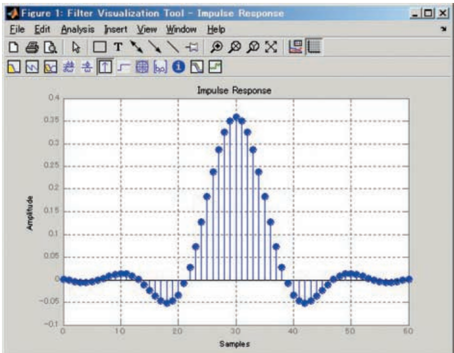


図2　設計したフィルタのインパルス応答

「fvtool (num,den)」と入力すると，設計したフィルタの特性を確認できる．

の設計を例に，System GeneratorによるモデリングからFPGAへの実装までの手順を説明していきます．

簡単なディジタル変復調システムの一例として，QPSK（quadrature phase shift keying ；直交位相シフト変調）のサンプル・モデルを利用します．ここではメッセージ信号の周波数帯域を抑圧する目的で，Raised Cosineロールオフ・フィルタを作成します．

まず，MATLABのコマンド・プロンプトからCommunication Toolboxのrcosineという関数を実行し，余弦ロールオフ・フィルタを作ります．

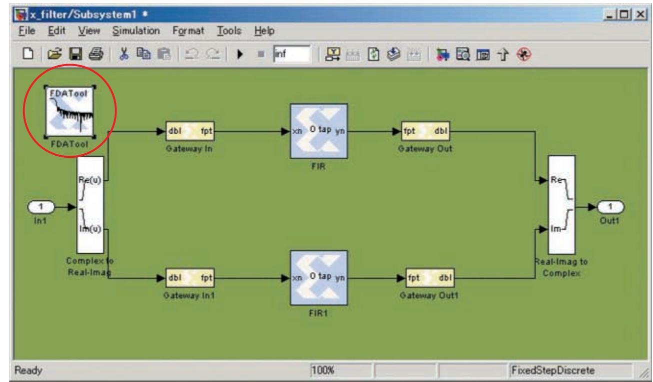
```
>> [num, den] = rcosine
          (1, 10,' fir /sqrt ', 0.5, 3) ;
```

上記のコマンドによって設計されるフィルタは，シンボルのサンプリング・レートとフィルタのサンプリング・レートの比が1：10，フィルタの構造がFIR/Root Cosineロールオフ，ロールオフ係数が0.5，群遅延が3となります．コマンドの詳細はhelpコマンドを参照してください．設計したフィルタの特性を確認するには，以下のコマンドを入力します（**図2**）．

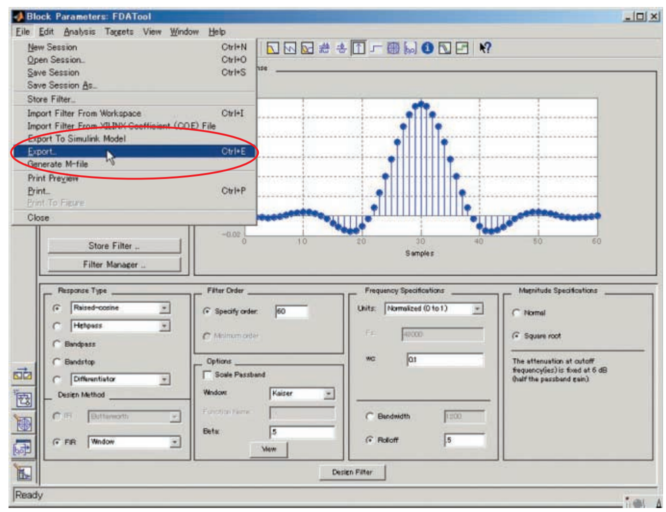
```
>> fvtool (num,den)
```

ディジタル・フィルタの係数は，その差分方程式の分子係数がnum，分母係数がdenという変数に格納されています．コマンド・ラインから以下のように変数名を入力して，値を確認します．

```
>> num
>> den
```



（**a**）FDAtoolブロックを配置



（**b**）フィルタ係数データの保存

図3　FDAToolによるフィルタ設計

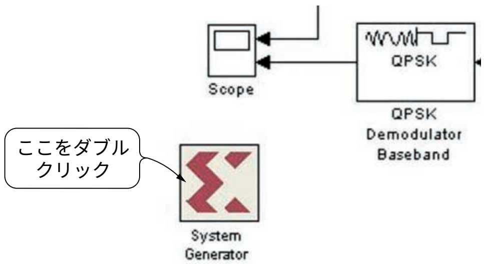
System Generatorは，フィルタ設計用のFDAToolを呼び出す「FDAtoolブロック」を用意している．

この例題で設計したディジタル・フィルタはFIRなので，分母係数denは1になります．

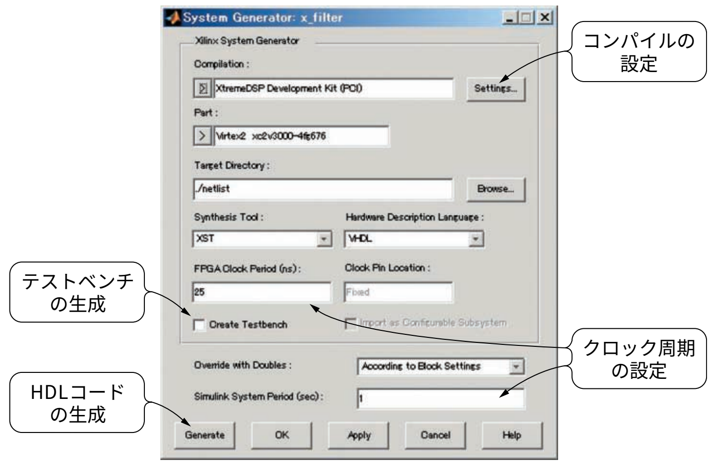
MATLABのオプション・ツールであるFilter Design Toolboxには，フィルタ設計用のGUI（graphical user interface）である「FDATool」があります．これに対してSystem Generatorは，このFDAToolを呼び出す「FDAtoolブロック」を用意しています．このブロックをモデル内に配置すると（**図3(a)**），ダイアログボックスの設定を行うだけでフィルタを設計できるようになります．また，メニューの「File」→「Export...」を選択すると，設計されたフィルタの係数データをMATLABのワーク・スペースに保存できます（**図3(b)**）．

● SimulinkとXilinx Blocksetを使ってモデリング

Simulink環境を起動し，QPSK送受信システム・モデルを設



（a）System Generatorブロックを配置



（b）ダイアログ

図4
System Generatorブロックとそのダイアログ
(a)のように，System Generatorブロックをモデル内に一つ配置する．このブロックをダブルクリックすると，(b)のような設定用のダイアログが開く．

定します．SimulinkではDiscrete Filterブロックを用いてディジタル・フィルタを表現します．

Simulinkを起動するには，以下のコマンドを実行します．

```
>> simulink
```

このコマンドにより，Simulinkライブラリ・ブラウザが起動します．System Generatorが提供するブロックは，このウィンドウから参照します．

次に，ライブラリ・ブラウザの［新規モデルの作成］ボタンを押すと，モデル設定ウィンドウが開きます．今回は誌面のつごう上，すでに設計されているモデルを読み込むだけとします．本誌付属のDVD-ROMに収録されているx_filter_original.mdlというファイルを選択してください．

モデルの中のRC Filterという名まえのサブシステムをダブルクリックすると，内部にDiscrete Filterというブロックがあります．これがディジタル・フィルタです．このブロックをダブルクリックして，フィルタの係数がMATLAB上で計算した値に設定されていることを確認してください．

続いて，シミュレーションを実行して，モデルの動作を確認します．パルス波形がロールオフ・フィルタによって整形されますが，シンボルの判定点（立ち上がりエッジ）ではメッセージ信号であるパルスと整形された波形の値が一致しているので，受信側では正しく復調されます．モデルに問題がなければ，Xilinx Blocksetを利用してFPGAに実装する部分の設計に入ります．

System Generatorを使ってモデリングを行う場合，モデルに以下の三つの要素を追加する必要があります．

- **System Generatorブロック**――HDL生成時の各種設定，シミュレーション時のFPGA Clock PeriodとSimulink System Periodの関連付け
- **Xilinx ブロック**――所望のアルゴリズムを表現するための機能ブロック
- **Gateway In/Outブロック**――System Generatorブロックと既存のSimulinkブロックのインターフェース

現在のモデルを別名で保存し（例えばx_filter1.mdl），図4(a)のようにSystem Generatorブロックをモデル内に一つ配置します．このブロックをダブルクリックすると，図4(b)のようなダイアログが開きます．

ここでFPGA Clock Period (ns)の欄に「25」を，Simulink System Period（sec）の欄に「1」を設定します．これにより，Simulink中のブロックが持つサンプル時間など，シミュレーションを行う際の最小ステップ幅とFPGAにシステムを実装する際に参照されるクロック時間が関連付けられます．つまり，Simulinkの最小ステップ幅である1がFPGAのクロック周期である25nsに設定されます．

本稿の例題では，受信側のロールオフ・フィルタをFPGAに実装していきます．まず，受信側のロールオフ・フィルタをXilinx BlocksetのFIRというブロックで表現します．このブロックは，Xilinx Blocksetの下のDSPというフォルダに格納されています．FIRブロックのパラメータ設定画面から，Discrete Filter ブロックの場合と同じようにフィルタ係数を変数として定義します．

パラメータ設定画面は，LatencyやSample Periodといった System Generator特有のパラメータ設定を行うダイアログ・ボックスを持っています．シミュレーションによってアルゴリズムを検討するという目的だけであれば，Latencyに「0」を与えて

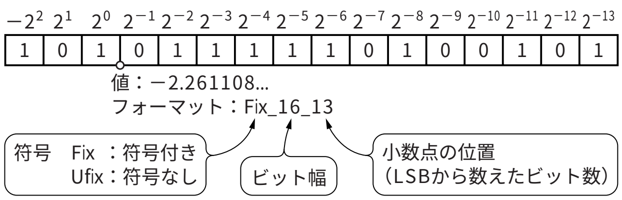


図5　System Generatorによる固定小数点表現の概念

この設定ではデータ・タイプが2の補数表現で，全体の語長が16ビット，小数部が13ビットとなる．

もかまいません．一方，FPGAに実装する場合は適切なレイテンシの値を与える必要があります．ここでは一例として「60」という値を入力します．

また，入出力信号を比較するため，Scopeブロックの前段にInteger Delayブロックを使用しています．ここに与える遅延量（シンボル遅延）は以下の式で与えられます．

遅延量（シンボル遅延）
　＝ロールオフ・フィルタによる群遅延×2（送受信）
　　＋FIRブロックのレイテンシ/レート変換率
　＝3×2＋60/10＝12

Sample Periodには，System Generator ブロックで設定したSimulink Periodに対する速度の比を与えます．最速の設定は「1」です．

SimulinkモデルとFPGAモデルのインターフェースを明示的に表現するため，Gateway In/Gateway Outブロックを挿入します．このインターフェース・ブロックは，System Generatorを使った設計ではかならず必要になります．

また，Simulink環境ではデータ・タイプをデフォルトで64ビット倍精度浮動小数点（double型）として扱いますが，FPGA設計では通常，有限語長の固定小数点としてシステムを記述する必要があります．Gateway In/Gateway Outブロックが必要になるのはそのためです．

● Simulinkモデルは浮動小数点，FPGAは固定小数点

上述のように，FPGAに実装するためには浮動小数点モデルを固定小数点モデルに変える必要があります．ここでは量子化に伴う注意点などについて言及しませんが，実際の設計ではシミュレーションによって量子化の影響を十分に検討する必要があります．量子化によって生じるおもな問題として，オーバフロー，アンダフロー，リミット・サイクル発振などが挙げられます．

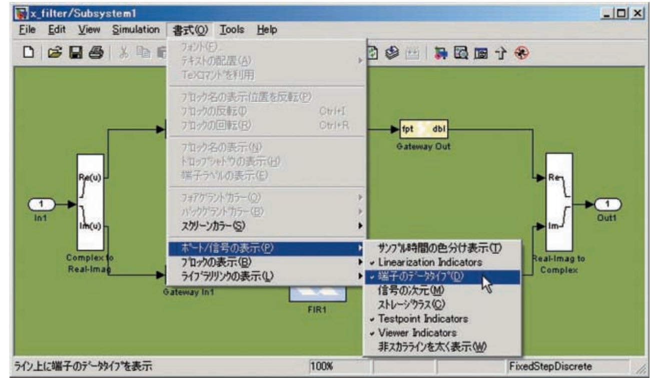


図6　データ・タイプの表示方法

Simulinkのモデル・ウィンドウから，「書式」→「ポート/信号の表示」→「端子のデータタイプ」を選択してシミュレーションを実行すると，信号線上にデータ・タイプが表示される．

固定小数点モデルをSystem Generatorで表現する場合，Simulinkブロックとのインターフェースである Gateway Inブロックに，データのビット長と小数点の位置をパラメータとして与えます．これで，固定小数点のデータ・タイプを利用できます．例えば，モデル中のGateway Inブロックの持つパラメータOutput Data Typeに「Signed（2 ’ comp）」，Numbers of Bitsに「8」，Binary Pointに「6」がすでに設定されています．この設定は，データ・タイプは2の補数表現で，全体の語長は8ビット，小数部は6ビットということを表しています．

図5に固定小数点表現の概念を示します．図の設定ではデータ・タイプが2の補数表現で，全体の語長は16ビット，小数部は13ビットとなります．

モデルで設定されているデータ・タイプを確認するには，Simulinkのモデル・ウィンドウから，「書式」→「ポート/信号の表示」→「端子のデータタイプ」を選択して，シミュレーションを実行します．これで，信号線上にデータ・タイプが表示されます（**図6**）．

● HDLとテストベンチを生成

System Generatorは，Xilinxブロックで作成されたモデルから等価なHDLコードを生成します．コードの生成過程は，モデルとSystem Generatorブロックの設定によって決定されます．また，この過程ではHDL以外のファイルも同時に生成されます．このファイルにはISE用のプロジェクト・ファイルや制約条件ファイルも含まれます．

System Generatorで設計，シミュレーション，コード生成を行う際には，かならずSystem Generatorブロックがモデル中に

含まれている必要があります．System Generatorブロックをダブルクリックしてダイアログを表示し，コード生成過程の設定を行います．

Compilationダイアログ・ボックスではコンパイル・タイプを指定します．通常は「HDL Netlist」が選択されます．このタイプが選択されると，HDLデータ，EDIFネットリスト，そのほかの補助的なファイルが生成されます．設定後，［Generate］ボタンを押すと，コード生成が始まります．

生成されたデザイン・ファイル（今回の例ではx_filter_files.vhd）にはclock，clock enableポートの記述があります．System Generatorはクロック・ラッパ・ファイル（x_filter_clk_wrapper.vhd）を生成します．各ファイルの関係を**図7**に示します．

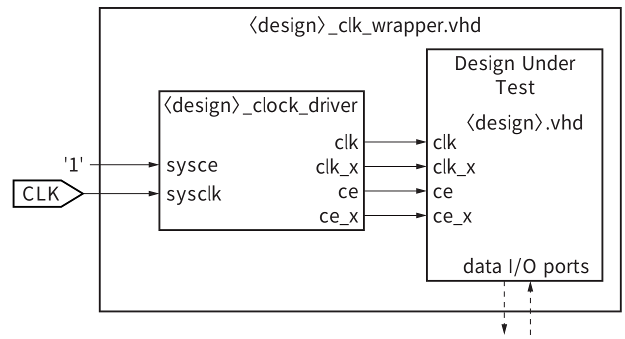


図7　生成されるデザイン・ファイルとクロック・ラッパ・ファイルの関係
System Generatorは，デザイン・ファイルといっしょにクロック・ラッパ・ファイルを生成する．

System Generatorによって作成されたモデルは，通常，Simulink上でシミュレーションされます．モデルによって設定されたビット長やクロック周期はFPGAに実装される設計と一致しますが，HDLシミュレータを使って詳細な動作を確認することもできます．System Generatorブロックをダブルクリックしてダイアログを開き，Create Testbenchチェックボックスを ONにしておくと，HDLとともにテストベンチが生成されます．

● ビットストリーム生成の方法は2通りある

System Generatorを利用してFPGAを開発する場合，ビットストリーム・データ（コンフィグレーション・データ）を作成する方法は，以下の2通りあります．

- **System GeneratorでHDLデータを生成した後，同時に生成されるプロジェクト・ファイルをISEで開き，論理合成以降の処理を行う方法**
- **System Generatorブロックからビットストリームを直接生成する方法**

それぞれの方法について，順番に説明していきます．

1）ISEを使って論理合成以降を処理する方法

まず，ISEを利用する方法から説明します．最初にSystem Generatorブロックをダブルクリックしてダイアログを開きます．ここでは一例として，Compilationを「HDL Netlist」に，Partを「Virtex2 xc2v3000-6ff1152」に，Target Directoryを「./netlist」に設定して，［Generate］ボタンを押します．これによって，HDLとともにISE用のプロジェクト・ファイルであるx_filter_clk_wrapper.nplが，MATLABのカレント・ディレクトリの下のnetlistディレクトリに生成されます．

次に，ISEを起動してこのプロジェクト・ファイルを開きます．デスクトップのProject Navigatorというアイコンをダブルクリックし，ISEの基本環境を開きます．「File」→「Open Project」を選択し，System Generatorが生成したファイルを格納しているディレクトリ（例えば「c:\matlab7\work\lab_sysgen\netlist」）のx_filter_clk_wrapper.nplを選択します．プロジェクトが表示されるので，左上に表示されているウィンドウのModule Viewタブを選択し，プロジェクトで管理されているファイル群のツリー構造をダブルクリックして展開します（**図8**）．

x_filter_clk_wrapper.vhdを選択し，左下のウィンドウのProcess Viewタブで表示される「Generate Program File」のところをダブルクリックしてください．これによって，System Generator上で設定された情報をもとに論理合成，配置配線が行われ，回路をFPGAに実装するためのビットストリーム・データが生成されます．

2）System Generatorでビットストリームを生成する方法

次に，System Generatorによってビットストリーム・データを直接生成する方法について説明します．これは**1）**より簡便な方法で，MATLABからISEに環境を移行することなく，一連の工程をすべてバッチ処理として実行します．System Generatorブロックをクリックしてダイアログを開き，Compilationを「bitstream」に変更してください．［Generate］ボタンを押すと，HDLコードの生成過程に続き，論理合成，配置配線が行われ，ビットストリーム・データが生成されます．

● 協調シミュレーションを利用して効率的にデバッグ

最後に協調シミュレーションについて説明します．

System Generatorによる協調シミュレーションをサポートしているFPGAボード（例えば，米国Nallatech社のDSP開発キットなど）をターゲットにSystem Generatorブロックのパラメー

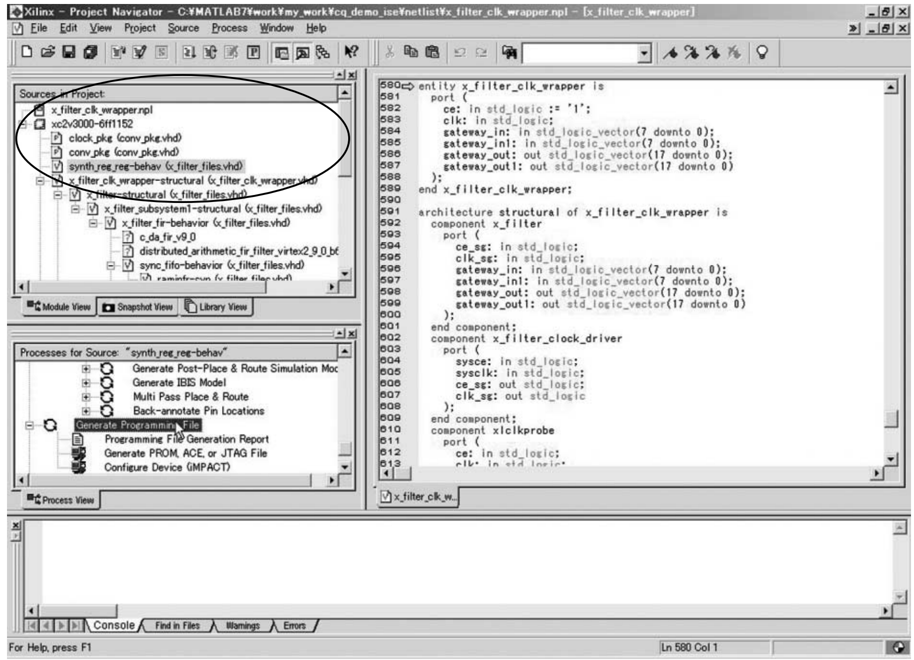
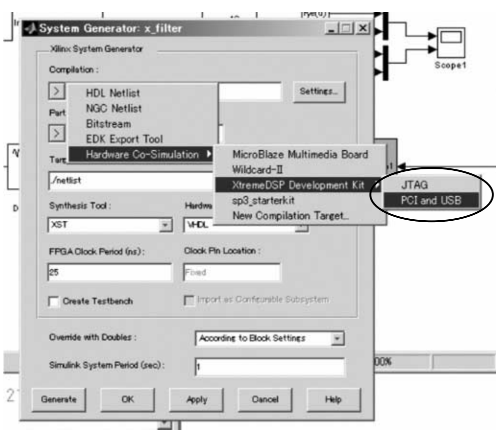
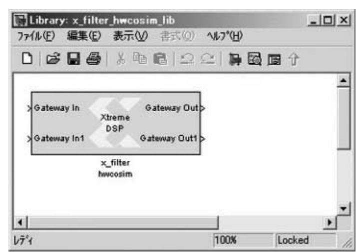


図8
Project Navigatorによるビットストリーム・データの生成
プロジェクトが表示されたら，左上に表示されているウィンドウのModule Viewタブを選択し，プロジェクトで管理されているファイル群のツリー構造をダブルクリックして展開する．



(**a**) ターゲットの選択



(**b**) インターフェース・ブロック

図9　協調シミュレーションの選択とインターフェース・ブロックの生成
System Generatorブロックのダイアログ上で，(a)のようにして「Xtreme DSP Kit」を選択する．[Generate]ボタンを押すと，ビットストリーム・データといっしょに(b)のような協調シミュレーション用インターフェース・ブロックが生成される．

タを設定すると，ビットストリーム・データといっしょに，これに関連付けられた協調シミュレーション用インターフェース・ブロックが生成されます．このインターフェース・ブロックをモデルにコピーし，シミュレーションすると，ターゲット・ハードウェア(FPGAボード)とホスト・パソコン(PC)の間のインターフェースが確立され，協調シミュレーションを実現できます．

まず，System Generatorブロックのダイアログからターゲットを選択します．ここでは一例として，Nallatech社のDSP開発キットである「Xtreme DSP Kit」を選択します(**図9(a)**)．[Generate]ボタンを押すと，ビットストリーム・データといっしょにインターフェース・ブロックが生成されます(**図9(b)**)．生成されたブロックをコピーし，元のモデルに挿入して結線します(**図10**)．なお，本誌付属のDVD-ROMには，すでに結線済みのx_filter_cosimというモデルも収録しています．元のモデルとの違いを確認してみてください．

このブロックは，シミュレーション時にほかのSimulinkブロックと同じように評価されますが，実際の処理はFPGA上で行われています．ブロックで扱われるデータ・タイプはSystem Generatorの設定と同じです．ブロックに入力されるデータがFPGAに伝播し，FPGA内部で処理された結果がブロックの出力としてSimulinkに渡されます．シミュレーションを実行する



図11
Scopeウィンドウによる信号表示
シミュレーション結果は，Simulinkの各種出力表示ブロックを使って確認できる．

と，その結果をSimulinkが提供する各種出力表示ブロックを使って確認できます(**図11**)．

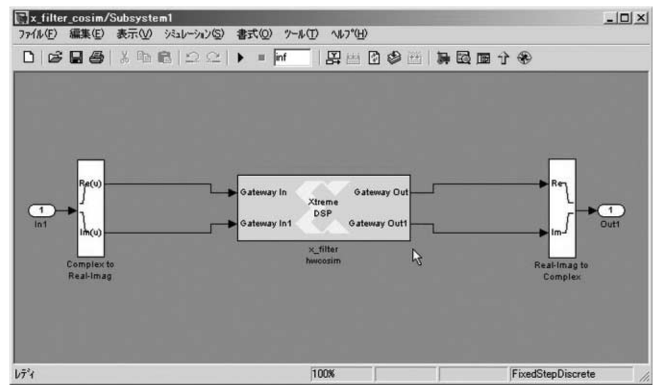


図10　インターフェース・ブロックのコピーと接続
System Generatorによって生成された協調シミュレーション用インターフェース・ブロックをコピーし，元のモデルに挿入して結線する．

もりた・やすし
ザイリンクス(株) マーケティング部 DSPソリューションマネージャー

<筆者プロフィール>
森田 康．1997年，東京理科大学大学院 工学研究科 電気工学専攻 修了．同年，日本テキサス・インスツルメンツ入社．バーンイン・テスト・ソケットの設計・開発に従事．2000年，サイバネットシステム入社．The MathWorks社の｢MATLAB / Simulink｣によるディジタル信号処理，通信システム関連ツール，Lyrtech社製信号処理システム開発ボードを担当．2004年，ザイリンクス入社．現在，DSPソリューションを担当．